ปาสคาลกับซี

บุญเลิศ เอี่ยมทัศนา

หากจะมีการจัดอันดับภาษาคอมพิวเตอร์เช่นเดียวกับการจัดอันดับเพลง ภาษาคอมพิวเตอร์ที่ท็อปฮิตติดอันดับโลกในปัจจุบันน่าจะได้แก่ ภาษาโปรล็อก(Prolog) ภาษาซีและภาษาปาสคาล(Pascal) ซึ่งภาษาโปรล็อกกล่าวกันว่าจะเป็นภาษาสำหรับคอมพิวเตอร์ในยุคที่ห้า สำหรับภาษาซีดูได้จากประกาศรับสมัครบุคคล ในประเทศไทยต้องการพนักงานพิมพ์ดีดผู้มีความรู้ภาษาอังกฤษ แต่สหรัฐต้องการวิศวกรผู้มีความรู้ในภาษาซี ส่วนปาสคาลเป็นภาษาสำหรับฝึกเขียนโปรแกรมและมหาวิทยาลัยในสหรัฐส่วนมากได้เปลี่ยนมาใช้ภาษาปาสคาลแทนฟอร์แทรน

ภาษาโปรล็อกและลิส์ป(LISP) เป็นภาษาของยุคใหม่โดยแท้ รูปแบบและกรรมวิธีในการเขียนโปรแกรมได้แตกต่างไปจากภาษาที่เราเคยรู้จักทั้งหมดอย่างสิ้นเชิง และผมยังไม่คุ้นกับมันจึงของดกล่าว ส่วนภาษาปาสคาลกับภาษาซีคล้ายกันมาก และสามารถใช้ได้บนเครื่องไมโครคอมพิวเตอร์เช่น ไอบีเอ็มตลอดไปจนถึงเครื่องคอมพิวเตอร์ขนาดใหญ่ จึงเป็นเรื่องที่น่าจะศึกษาเปรียบเทียบกัน

ลักษณะโดยทั่วไป

ภาษาทั้งสองนี้ผู้สร้างได้สร้างด้วยจุดมุ่งหมายที่ต่างกันคือ ภาษาปาสคาลสำหรับเพื่อการเรียนการสอนเขียนโปรแกรม ส่วนภาษาซีเพื่อเป็นเครื่องมือในการเขียนระบบจัดการ UNIX เสร็จแล้วจึงใช้เป็นเครื่องมือในการเขียนโปรแกรมของนักเขียนโปรแกรมระดับระบบจัดการ(system programmer)

คำว่าเครื่องมือออกจะเข้าใจยาก หากคุณเคยเขียนโปรแกรมในภาษาเครื่องคงจะพอนึกออกว่า เมื่อเริ่มต้นเขียนเราจะรู้สึกอึดอัดเพราะไม่มีเครื่องมือใดเลย เมื่อเขียนไปสร้างโปรแกรมย่อยไปโปรแกรมย่อยนี้จะเป็นเครื่องมือในการเขียนโปรแกรมส่วนอื่นๆ ต่อไปได้สะดวก ผู้เขียนโปรแกรมระบบจัดการ UNIX ก็ได้ใช้วิธีนี้แต่ได้ดัดแปลงเป็นภาษาอีกหนึ่งภาษาเรียกว่า ภาษาซี

ทั้งสองภาษานี้ถือว่าเป็นภาษาในระดับสูง(high level language) แต่ซีเป็นภาษาสูงในระดับที่ค่อนข้างต่ำ คำที่ใช้ในภาษานี้จะหนักไปทางด้านการจัดการในเรื่องหน่วยความจำและอินพุต–เอาต์พุต และขาดสิ่งอำนวยความสะดวกบางประการที่ภาษาสูงควรจะมี เช่น เมื่อเทียบกับภาษาปาสคาลในการทำงานกับไฟล์ของเรคอร์ด เมื่อใช้ภาษาซีเราต้องเขียนโปรแกรมเพื่อดำเนินการเอง แต่ในภาษาปาสคาลมีคำสั่งให้เราใช้ได้โดยสะดวก

รูปแบบ

ถึงแม้ทั้งสองภาษานี้จะมีความมุ่งหมายในการสร้างแตกต่างกัน ระดับของภาษาแตก

```
/* SRTSTR.C */
   /* reads in string and sorts them */
#include <stdio.h>
#define LEN  81     /* max length of string */
#define LINE 20     /* max number of line to be read */
#define HALT ""     /* null string to stop input */
main()
{
  static char any[LINE][LEN];  /* array to store input */
  char *ptstr[LINE];           /* array of pointer variable */
  int  inc = 0;                  /* input count */
  int  cntr;                   /* output count */

  printf ("Input up to %d lines,and I will sort them.\n",LINE);
  printf ("To stop,hit the ENTER key at a line's start.\n");
  while (gets(any[inc]) != NULL && strcmp(any[inc],HALT)
                                && inc++ < LINE)
     ptstr[inc - 1] = any[inc - 1];

  Strsort(ptstr,inc);           /* string sorter */

  puts("\n Here's the sorted list:\n");
  for ( cntr = 0; cntr < inc; cntr++)
    puts(ptstr[cntr]);
}

/* string-pointer-sorting function */
Strsort(str,num)
char *str[];
int num;
{
  char *tem;
  int top,seek;

  for (top = 0; top < num - 1; top++)
    for (seek = top + 1; seek < num; seek++)
      if (strcmp(str[top],str[seek]) > 0 )
      {
        tem = str[top];
        str[top] = str[seek];
        str[seek] = tem;
      }
}
```

รูปที่ 1 โปรแกรมจัดเรียงลำดับข้อมูลจากคีย์บอร์ดในภาษาซี

```
{  SRTSTR.PAS  }
   (* read in string and sort them *)
PROGRAM  SortString(input,output);
CONST  len  = 80;     (* max length of string *)
       line = 20;     (* max number of line *)
       HALT = '';     (* null string *)
       inc  : integer = 1;
TYPE   strT = String[len];
       anyT = ARRAY[1..line] OF strT;
VAR    any  : anyT;                    (* arraay to store input *)
       str  : strT;                    (* input count *)
       cntr : integer;


{ Procedure to sort array of string }
PROCEDURE  SortStr(VAR strn : anyT; num : integer);
VAR  tem : strT;
     top,seek : integer;
BEGIN
  FOR top := 1 TO num - 1  DO
    FOR seek := top + 1 TO num DO
      IF strn[top] > strn[seek] THEN
        BEGIN
        tem := strn[top];
        strn[top] := strn[seek];
        strn[seek] := tem;
        END;
END;

BEGIN  { Main }
WriteLn('Input up to  ,line,  and I will sort them');
WriteLn('To stop,hit the ENTER key at a line''s start');
ReadLn(any[inc]);
WHILE ( any[inc] <> HALT ) AND ( inc < line ) DO
  BEGIN
  inc := inc + 1;
  ReadLn(any[inc]);
  END;

SortStr(any,inc);

WriteLn('Here is the sorted string');
FOR cntr := 1 TO inc DO
  WriteLn(any[cntr])
END.
B>
```

รูปที่ 2 โปรแกรมจัดเรียงลำดับข้อมูลจากคีย์บอร์ดในภาษาปาสคาล

ต่างกัน แต่ในการเขียนโปรแกรมใช้งานทั่วไปแล้วจะคล้ายกันมาก จนดูไม่ออกถึงระดับของภาษาที่แตกต่างกัน ขอให้ดูโปรแกรมในรูปที่ 1 และ 2 ทั้งสองโปรแกรมนี้ได้ให้ชื่อในดิสค์ไว้เหมือนกันคือ SRTSTR.PAS กับ SRTSTR.C ทั้งสองโปรแกรมนี้ทำงานเดียวกันคือ อ่านสตริงที่ผู้ใช้ป้อนทางคีย์บอร์ด เช่น อาจเป็นชื่อบุคคล จำนวนไม่เกิน 20 ชื่อ หรือเมื่อผู้ใช้กดคีย์ ENTER ที่ต้นบรรทัด เสร็จแล้วนำไปเรียงลำดับตามลำดับอักษร แล้วแสดงผลบนจอภาพดังแสดงในรูปที่ 3

จะเห็นว่าทั้ง 2 ภาษามีรูปแบบเหมือนกันคือ ไม่ใช้หมายเลขบรรทัด ใช้โปรแกรมย่อยในแบบสร้างคำใหม่ จึงทำให้เขียนโปรแกรมเป็นโมดูล ซึ่งในแต่ละโมดูลยังมีส่วนประกอบเหมือนกันอีก คือประกอบด้วย

- ส่วนหัว (heading)
- ส่วนข้อกำหนด (declaration part)
- ส่วนถ้อยแถลง (statement part)

มีข้อแตกต่างกันอยู่ 2 เรื่องใหญ่ๆ คือ การกำหนดสิ่งที่อยู่ในข้อกำหนดกับฐานะของโปรแกรมย่อย แต่ขอเรียนให้ทราบถึงความหมายของตัวโปรแกรมก่อน ขอให้ดูในรูปที่ 4 ซึ่งผมได้นำรูปแบบง่ายๆ ของโปรแกรมทั้งสองภาษามารวมกันไว้ ทั้งหมดนี้(ในแต่ละภาษา)เรียกว่าโปรแกรม แต่ถ้ากล่าวถึงตัวโปรแกรม(body) ในภาษาปาสคาลคือส่วนที่อยู่ระหว่างคำว่า BEGIN ถึง END. ในภาษาซีคือส่วนที่อยู่ระหว่างวงเล็บปีกกา

```
D>srtstr
Input up to 20 lines,and I will sort them.
To stop,hit the ENTER key at a line's start.
Pat Alazar
Ralph Destry
Peteer Howser
John Brown
Alice Peters
Jean Freitag

 Here's the sorted list:

Alice Peters
Jean Freitag
John Brown
Pat Alazar
Peteer Howser
Ralph Destry

D>
```

ป้อนทางคีย์บอร์ด

รูปที่ 3 ผลการรันโปรแกรมจัดเรียงลำดับข้อมูล รายชื่อบุคคล ท่อนบนคือที่ผู้ใช้ป้อน ท่อนล่างคือที่เครื่องจัดเรียงแล้วนำผลออกแสดงทางจอภาพ

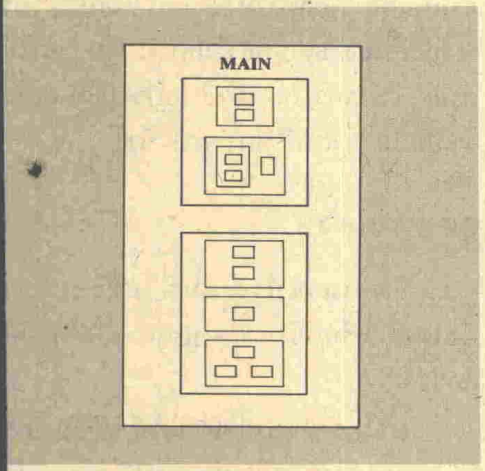
```
/* SIMPLE.C */
   /*  simple program in C  */
main()
{
  puts("This is a simple program in C.");
  puts("  C was created as a toal for working programmers.");
}

(* SIMPLE.PAS *)
   (*  simple program in Pascal  *)
PROGRAM  Simple(output);
BEGIN
  WriteLn('This is a simple program in Pascal');
  WriteLn('  Pascal was created as a toal for lern programming')
END.
```

รูปที่ 4 โปรแกรมง่ายๆ ในภาษาซีและปาสคาล

ส่วนข้อกำหนด

หากถือหลักของภาษาปาสคาลส่วนข้อกำหนดคือ การกำหนดค่าคงตัว(CONST) การกำหนดแบบของข้อมูล(TYPE) การกำหนดตัวแปร(VAR) และโปรแกรมย่อย (Procedure และ Function) ตามรูปที่ 2 จะเห็นว่าในภาษาปาสคาลกำหนดส่วนที่เป็นข้อกำหนดไว้ก่อนตัวโปรแกรม ส่วนในภาษาซีมีทั้งก่อน(ก่อนส่วนหัว) ใน และหลังตัวโปรแกรม

เมื่อนำมาเปรียบเทียบกันเว้นโปรแกรมย่อยที่จะกล่าวต่อไป ไม่น่าจะมีข้อได้เปรียบและเสียเปรียบกัน นอกจากในภาษาปาสคาลดูง่ายกว่าและมีชื่อกำกับไว้ด้วย



รูปที่ 5 การมองโปรแกรมในปาสคาลในลักษณะโมดูล แต่ละสี่เหลี่ยมคือโมดูลหนึ่งๆ

โปรแกรมย่อย

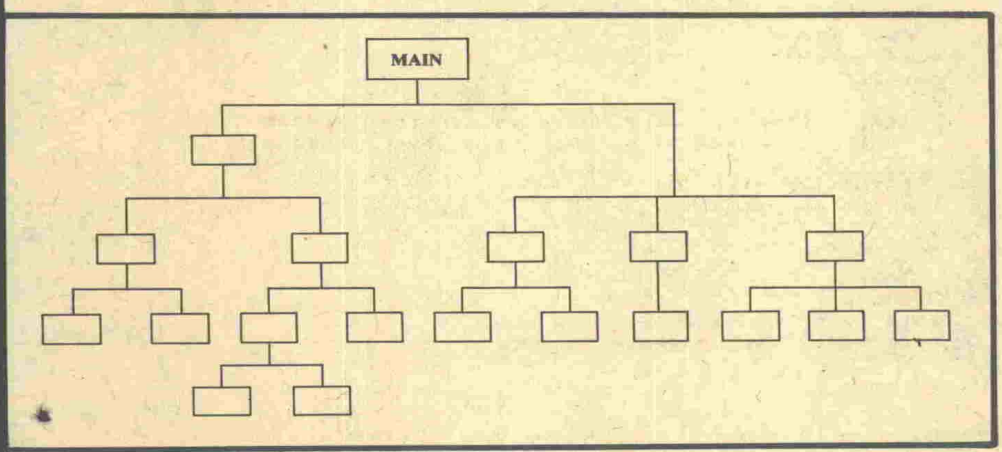
ในภาษาปาสคาลมีโปรแกรมหลักและโปรแกรมย่อย ซึ่งแบ่งออกเป็น 2 แบบคือ Procedure ซึ่งเปรียบเสมือนคำสั่ง และ Function ซึ่งเปรียบเสมือนตัวแปร แต่ในภาษาซีถือว่าโปรแกรมทุกโมดูลเป็นฟังก์ชันเหมือนกันหมด รวมทั้งโมดูลที่เรียกว่า main ( ) คือส่วนที่เป็นโปรแกรมหลักก็เป็นฟังก์ชัน แต่ฟังก์ชันในภาษาซีเป็นได้ทั้งโพรซิเดียวและฟังก์ชัน หากเราเป็นผู้สร้างขึ้นใหม่ขึ้นอยู่กับเราจะกำหนดให้เป็นโพรซิเดียวหรือฟังก์ชัน ตามโปรแกรมในรูปที่ 1 ได้กำหนดให้ Strsort เป็นโพรซิเดียว

ในภาษาปาสคาลโมดูลมีฐานะไม่เท่ากัน คือ เมื่อเราเขียนโปรแกรมหลักที่มีโปรแกรมย่อย โปรแกรมย่อยจะอยู่ในโมดูลของโปรแกรมหลัก และถ้าในโปรแกรมย่อยมีโปรแกรมย่อยรองลงไปอีก โปรแกรมย่อยรองลงไปนี้ก็จะอยู่ในโมดูลของโปรแกรมย่อยนั้น และจะรอง ๆ ลงไปกี่ชั้นก็ได้ดังแสดงในรูปที่ 5 เมื่อมองในแบบโมดูลรองในโมดูลใหญ่ หรือจะมองแบบความสัมพันธ์ได้ดังรูปที่ 6

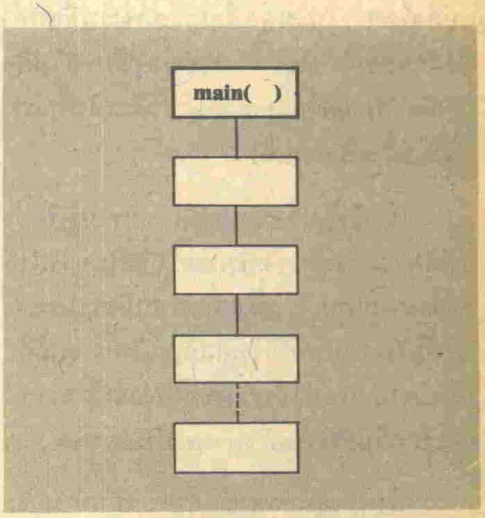
ส่วนภาษาซีถือว่าโมดูลทุกโมดูลมีฐานะเท่าเทียมกันหมดรวมทั้งmain( ) ด้วย มีข้อกำหนดแต่เพียงว่าให้เขียน main( ) ขึ้นต้นโปรแกรมย่อยหรือโมดูลนอกนั้นให้เขียนต่อท้าย ลักษณะของโปรแกรมในภาษาซีจึงเป็นดังรูปที่ 7 แต่ที่แสดงในรูปไม่ได้คือความสัมพันธ์ของแต่ละโมดูล โดยที่ถือว่าทุกโมดูลมีฐานะเท่ากันโปรแกรมย่อยจึงสามารถเรียกใช้ main( ) ได้ และจะเรียกใช้โปรแกรมย่อยอื่นใดก็ได้อย่างเสรี ผิดกับภาษาปาสคาลที่เรียกใช้ได้เฉพาะโปรแกรมย่อยที่อยู่ในโมดูลของตนหรือที่อยู่ก่อนตนเท่านั้น

เมื่อเป็นเช่นนี้จะเห็นว่าการเขียนโปรแกรมในภาษาซีใกล้เคียงกับวิธีการเขียนโปรแกรมในภาษาเครื่องมาก แต่ง่ายกว่าภาษาเครื่องและสะดวกกว่าภาษาปาสคาล เพราะสามารถสร้างคำใหม่และเขียนขยายความ (stepwise refinement)ไปได้เรื่อยๆ เมื่อป้อนก็ป้อนตรงตามที่เขียนไว้ ต่างกับภาษาปาสคาลซึ่งเขียนเป็นขยายความเช่นกัน แต่เวลาป้อนต้องนำสิ่งที่เขียนทีหลังมาป้อนก่อน และการเขียนโปรแกรมเป็นโมดูลซ้อนกันหลายๆ ชั้นบางครั้งก็ทำให้อ่านโปรแกรมยาก

นอกจากจะเขียนโปรแกรมย่อยเป็นภาษาของตนแล้ว ทั้งปาสคาลและซียังสามารถเรียกใช้โปรแกรมย่อยที่เป็นภาษาเครื่องได้



รูปที่ 6 แสดงความสัมพันธ์ของโปรแกรมในภาษาปาสคาล



รูปที่ 7 ลักษณะของโปรแกรมในภาษาซี(รูปนี้ไม่ได้แสดงความสัมพันธ์ของโมดูลดังเช่นรูปที่ 6 หากแสดงถึงรูปแบบของโปรแกรมในภาษาซีซึ่งแต่ละโมดูลจะเรียงต่อกันไป ไม่ซ้อนกันเช่น ปาสคาล

การผ่านค่า

ทั้งภาษาปาสคาลและภาษาซีกำหนดให้มีการผ่านค่าระหว่างโมดูลที่เรียกกับโมดูลที่ถูกเรียกไว้เหมือนกันคือ

– ไม่มีการผ่านค่า

– ใช้ตัวแปรที่เป็นโกลบอล (global variable)

– ผ่านค่าทางเดียว (pass by value)

– ผ่านค่าไปกลับ (pass by reference)

โปรแกรมย่อยที่ไม่มีการผ่านค่าไปให้ได้แก่โปรแกรมย่อยที่ทำงานโดยไม่เกี่ยวกับตัวแปร เช่น ClrScr คือให้ลบจอภาพในภาษาปาสคาล ส่วนการใช้ตัวแปรที่เป็นโกลบอลคือที่กำหนดไว้เป็นส่วนกลางซึ่งโมดูลใดจะใช้ก็ได้ การส่งค่าด้วยวิธีนี้ไม่สมควรใช้ในการเขียนโปรแกรมเป็นโมดูลคือ ทั้งในภาษาปาสคาลและซี

สำหรับการผ่านค่าทางเดียวเป็นการผ่านค่าจากโมดูลที่เรียกไปยังโมดูลที่ถูกเรียก การเปลี่ยนค่าใด ๆ ในโมดูลที่ถูกเรียกจะไม่เกิดผลแก่ตัวแปรเดิม เพราะเพียงผ่านแต่ค่าไปให้เท่านั้น (จึงเรียกว่า pass by value) ส่วนการผ่านค่าไปกลับหากเกิดการเปลี่ยนแปลงใดๆ ในโมดูลที่ถูกเรียกจะเปลี่ยนค่าในตัวแปรเดิมด้วย เพราะค่าที่ผ่านไปคือแอดเดรสของข้อมูลของตัวแปรเดิม(จึงเรียกว่า pass by reference)

เมื่อใช้ในภาษาปาสคาลเราไม่รู้สึกยุ่งยากเพราะคงผ่านชื่อตัวแปรไปให้ในการผ่านค่าทั้งสองแบบ การจำแนกค่าเป็นการผ่านค่าแบบใดจะกำหนดในโมดูลที่ถูกเรียก ข้อดีคือใช้สะดวก ข้อเสียคือเราจะต้องจดจำว่าโมดูลที่เราเรียกไปนั้นรับการผ่านค่าในแบบใด

สำหรับในภาษาซีกลับตรงกันข้าม จะต้องกำหนดแบบของการผ่านค่าไว้ในโมดูลที่เรียกคือ ถ้าผ่านทางเดียวให้ผ่านชื่อตัวแปร ถ้าผ่านไปกลับให้ผ่านแอดเดรสของตัวแปร แต่ถ้าตัวแปรเป็นแบบสตริงหรือเป็นแบบมีโครงสร้างจะต้องผ่านแอดเดรส นั่นคือจะเป็นการผ่านค่าไปกลับเสมอ ทำให้ต้องระวังในการเขียนโปรแกรม นอกจากนี้ชื่อตัวแปรมาตรฐานเช่นตัวแปรแบบเลขจำนวนเต็มจะเป็นตัวแปรในนัยของคณิตศาสตร์ ถ้าจะให้หมายถึงแอดเดรสต้องนำด้วย & แต่ชื่อตัวแปรแบบสตริงและที่เป็นโครงสร้างจะหมายถึงแอดเดรส ถ้าจะให้หมายถึงข้อมูลให้นำด้วย * และยังมีเรื่องปลีกย่อยอีก ทำให้การใช้ตัวแปรในภาษาซีค่อนข้างจะยุ่งยากสำหรับผู้เริ่มศึกษา

คำในภาษา

ทั้งในภาษาปาสคาลและซีแบ่งคำออกเป็น 3 พวกเหมือนกัน แต่เรียกชื่อแตกต่างกันคือ

– คำที่ใช้ในการควบคุมการทำงานของโปรแกรม เช่น FOR, WHILE, CHAR ในภาษาปาสคาลเรียก reserved words ในภาษาซีเรียก keywords

– คำที่เป็นโปรแกรมย่อย ในภาษาปาสคาลคือทั้งที่เป็นโพรซิเดียวและฟังก์ชัน เรียกว่า standard identifiers ซึ่งในกลุ่มคำเหล่านี้ได้รวมคำที่เป็นค่าคงตัวเช่น Pi ไว้ด้วย ส่วนในภาษาซีคือที่เป็นฟังก์ชันเรียกว่า library functions และยังหมายรวมถึงคำที่เป็น macro ด้วย

– คำที่ผู้ใช้คือผู้เขียนโปรแกรมกำหนดขึ้นใหม่ได้แก่ แบบข้อมูล ค่าคงตัว ตัวแปร โปรแกรมย่อย และเฉพาะในซีคือมาโคร

คำที่ผมจะนำมากล่าวถึงคือ คำที่เป็นโปรแกรมย่อย ซึ่งในภาษาซีเรียกว่า ฟังก์ชัน และค่อนข้างจะสับสนพอสมควร เพราะบางคำเป็นฟังก์ชันแท้ๆ บางคำใช้ได้ทั้งในรูปของโพรซิเดียวและฟังก์ชัน(ค่าที่ส่งกลับมาส่วนมากแสดงความสำเร็จหรือไม่สำเร็จของการปฏิบัติตามคำสั่งนั้นๆ) บางคำซึ่งเป็นส่วนน้อยเป็นโพรซิเดียวแท้ๆ

นอกจากนี้คำในภาษาซียังมีหลายระดับคือ เกี่ยวกับการจัดหน่วยความจำแบ่งเป็น 3 ระดับ เกี่ยวกับอินพุตเอาต์พุตแบ่งเป็น 2 ระดับ(ซึ่งฟังก์ชันในระดับสูงจะเรียกใช้ฟังก์ชันในระดับต่ำ) ข้อดีคือ เราสามารถเลือกใช้ฟังก์ชันในระดับที่เราต้องการได้ เป็นการลดจำนวนอ็อปโค้ดทำให้โปรแกรมที่เป็นภาษาเครื่องสั้นลง แต่เมื่อยังไม่คุ้นจะเห็นว่าคำในภาษาซีมีมากกว่าในภาษาปาสคาลจนเกินความจำเป็นและยุ่งยากในการนำมาใช้

จุดเด่นของซี

ในภาษาซีมีจุดเด่นที่ไม่มีในภาษาปาสคาล หรือที่มีในภาษาปาสคาลแต่ซีทำได้ดีกว่าคือ

มาโคร ในภาษาซียอมให้เราใช้มาโครได้ดังแสดงในโปรแกรมตามรูปที่ 8 มาโครต่างกับโปรแกรมย่อยตรงที่ ทุกแห่งที่เราใช้

```
/* MACRO.C */
   /*  macro in C  */
#define PRINT(X) printf("%s to you.\n",X)
main()
{
  static char *first = "This is my first message";
  static char *scnd  = "this is my last message";

  PRINT(first);
  PRINT(scnd);
}
B>

D>macro
This is my first message to you.
this is my last message to you.

D>
```

รูปที่ 8 แสดงการใช้มาโครในภาษาซีคำว่า PRINT(X) คือมาโคร

าโครในโปรแกรม คอมพายเลอร์จะย้าย วนของโปรแกรมที่เป็นมาโครมาไว้ตรง เราใช้ แต่ถ้าเป็นการใช้โปรแกรมย่อยจะเป็น ารเรียกไปยังโปรแกรมย่อยนั้น การใช้มาโคร งเปลืองเนื้อที่ในหน่วยความจำแต่ประหยัด วลา จึงขึ้นอยู่กับผู้เขียนโปรแกรมจะเลือกใช้

อ๊อปโค๊ด จากโปรแกรมตัวอย่างใน ปที่ 1 และที่ 2 เมื่อคอมพายล์แล้วแต่ยังไม่ าโปรแกรมย่อยมารวม โปรแกรมภาษาเครื่อง องภาษาซีจะสั้นกว่าภาษาปาสคาลมากคือ าวเพียง 299 ไบท์ ส่วนของปาสคาลยาวถึง 40 ไบท์ ในด้านความเร็วผลการรันโปรแกรม ามรูปที่ 9 ปรากฏว่าไมโครซอฟท์ซีใช้เวลา 8 วินาที เทอร์โบปาสคาลเวอร์ชั่น 3 ใช้เวลา 2 วินาที ผลการทดลองนี้ยังไม่อาจตัดสิน ห้เด่นชัดได้ว่าซีเร็วกว่าปาสคาลเพียงแต่ ปรียบเทียบให้ดูเท่านั้น

การให้ค่าเริ่มต้น ในภาษาซีเราสามารถ ห้ค่าเริ่มต้นแก่ตัวแปรได้ดังที่ให้แก่ BELL ในภาษาปาสคาลมาตรฐานให้ไม่ได้ต้องเขียนในโปรแกรมซึ่งบางครั้งต้องเขียนโปรแกรมยาวมาก เช่น เมื่อสร้างต่ออาร์เรย์ชื่อเดือนในรอบปี ที่ตัวอย่างในรูปที่ 9 ให้ได้เพราะเป็นเทอร์โบปาสคาลแต่ก็ถือเป็นค่าคงตัว

นอกจากนี้ยังมีเรื่องปลีกย่อยอีก เช่น การกำหนดเงื่อนไข การเพิ่มหนึ่งและลดหนึ่งแก่ตัวแปรซึ่งภาษาซีเขียนได้สั้นกว่าปาสคาล

ศึกษาภาษาใดดี

จากที่กล่าวมานี้เป็นการเปรียบเทียบภาษาปาสคาลกับซีอย่างกว้างๆ ซึ่งคุณอาจจะเห็นว่าไม่แตกต่างกันนัก เพราะทั้งสองภาษานี้ใกล้เคียงกันมาก หากเปรียบกับภาษาฟอร์–แทรนจะเห็นฉนัดว่าภาษาซีใกล้กับภาษาเครื่อง เพราะภาษาเครื่องเป็นภาษาที่กว้างเขียนโปร–แกรมได้ทุกรูปแบบ(ขึ้นกับความสามารถของผู้เขียน) ตัวโปรแกรมมีขนาดเล็กและเมื่อรันจะใช้เวลาน้อยมาก

ส่วนภาษาในระดับสูงเป็นภาษาเฉพาะเรื่อง เช่น ฟอร์แทรนเหมาะสำหรับงานด้านคำนวณโคบอลเหมาะสำหรับงานบัญชี FORTH เหมาะสำหรับงานควบคุม เขียนโปรแกรมได้สั้นกว่า ง่ายกว่า และเร็วกว่าภาษาเครื่อง แต่เมื่อแปรเป็นภาษาเครื่องจะได้แมชีนโค้ดยาวกว่าทำให้ใช้เวลาในการรันมากกว่า

ที่ว่าซีใกล้เคียงกับภาษาเครื่องนั้นหมายถึง ในงานเขียนโปรแกรมระบบจัดการและหมายรวมตลอดไปถึงงานดำเนินกรรมวิธีข้อมูล แต่ในงานควบคุมอัตโนมัติผมไม่แน่ใจว่าจะใช้ซีได้อย่างเหมาะสมหรือไม่จึงขอตัดเรื่องนี้ออก งานดำเนินกรรมวิธีข้อมูลกล่าวอย่างง่ายๆ คืองานที่ใช้เครื่องคอมพิวเตอร์ส่วนบุคคล(แอปเปิ้ล, IBM PC) ทำอยู่ในปัจจุบัน ซึ่งรวมถึงกราฟิคและการสร้างภาพเคลื่อนไหวด้วย ปาสคาลดูจะด้อยกว่าภาษาซี ในด้านความเร็วแต่ก็เขียนโปรแกรมได้ง่ายกว่า ข้อผิดพลาดหากมีเกือบทั้งหมดจะพบเมื่อคอมพายล์ ส่วนภาษาซีพบเป็นส่วนมากเมื่อคอมพายล์แต่ก็มีไม่น้อยที่พบเมื่อรัน ไม่ใช่เป็นเพราะคอมพายเลอร์ หากเป็นเพราะรูปแบบของภาษาเอง

การที่จะศึกษาภาษาใดจึงขึ้นกับความต้องการของแต่ละบุคคล หากงานเขียนโปรแกรมของคุณต้องการความรู้ในภาษาหนึ่งภาษาใดอย่างลึกซึ้งขนาด system programmer ก็ควรศึกษาภาษาซี แต่ถ้าเป็นงานที่ต้องการความรู้ทางด้านภาษาเพียงพอรู้(ซึ่งส่วนมากมักเป็นงานในระดับนี้) ก็ควรศึกษาภาษาปาสคาล เพราะศึกษาได้ง่ายกว่า

แต่ถึงอย่างไรก็ตามการศึกษาภาษาซีโดยตรงเป็นเรื่องยากพอสมควร หากรู้ภาษาปาสคาลมาก่อนจะเข้าใจภาษาซีได้โดยง่าย และภาษาปาสคาลเป็นภาษาที่สร้างขึ้นมาเพื่อการเรียนการสอน เขียนโปรแกรม การที่จะศึกษาภาษาอีกหนึ่งภาษาไม่ใช่เรื่องยาก เรื่องที่ยากกว่าคือการเขียนโปรแกรม

```
{ BENCH.PAS }
  (*  benchmark for C and Pascal  *)

PROGRAM  Bench(output);
CONST  BELL = #7;
VAR    cntr,a,b : Integer;
BEGIN .
Write(BELL);
a := 2; b := 1;
FOR cntr := 1 TO 30000 DO
  IF a > b THEN
    a := a - 1
  ELSE
    a := a + 1;
Write(BELL);
END.
B> - - - - - - - - - - - - - - - -
/* BENCH.C */
   /*  benchmark for C and Pascal  */

#define BELL 7
main()
{
  int cntr,a,b;

  putch(BELL);
  a = 2; b =1;
  for ( cntr = 1; cntr < 30001; cntr++ )
    if ( a > b )
      a--;
    else
      a++;
  putch(BELL);
}
```

ปที่ 9 แสดงโปรแกรมในภาษาปาสคาลกับซีที่ใช้ในการเทียบความเร็ว